

Model Visualization - Shape Identification

Major Goal:

The primary aim of this feature is to develop an interactive tool within a Streamlit app that enables users to upload a 3D model (specifically in STL format), analyze its geometry, and automatically detect and annotate basic shapes. For the initial implementation, we focus on identifying circular faces within models, with the end goal of recognizing and displaying key geometrical attributes like diameter and circumference. This feature is intended to be a foundational component for more complex shape detection and feature extraction processes within 3D models, ultimately aiding in design, analysis, and manufacturing workflows.

What We Have Achieved So Far:

- Model Upload and Loading: Users can upload STL files, which are loaded and displayed using Plotly, providing a basic visualization of the 3D model.
- Basic Circular Face Identification Logic: Initial logic has been implemented to identify circular faces within the model by calculating the distances of vertex points from a centroid. If distances align closely within a tolerance, the face is tentatively identified as circular.
- Visualization Framework: Established a Plotly-based visualization structure to display the model and any identified circular faces with annotations, allowing hover-based display of calculated dimensions like diameter and circumference.

Challenges and Current Limitations:

- Surface Approximation in STL Models: STL files represent curved surfaces through multiple triangular faces rather than single geometric entities, which complicates the detection of circles in cylindrical shapes. Detecting circular features from a mesh of triangles requires advanced logic to cluster triangular faces or approximate curves.
- Tolerance Adjustment and Complex Models: Circular detection based on a fixed tolerance may miss approximations in real-world models, where faces aren't perfectly equidistant from a center. This limitation reduces reliability and highlights the need for more adaptive algorithms.
- Need for Enhanced Detection Algorithms: Simple distance-based methods are insufficient for accurate shape identification in 3D models with complex surfaces or curves. Advanced techniques, such as clustering algorithms or spatial analysis, are required to correctly detect and classify these features, especially in models with intricate geometries or small tolerances.

Next Steps and Additional Research Requirements:

Achieving robust shape identification requires deeper exploration and implementation of advanced computational geometry methods, possibly including:

- Enhanced Clustering for Approximate Shapes: Developing methods to cluster sets of triangles that approximate a circular face.
- Adaptable Tolerances and Multi-Layered Analysis: Adjusting detection thresholds dynamically based on model complexity and refining centroid calculations to improve accuracy.
- Algorithm Research and Testing: Investigating more sophisticated shape-recognition algorithms or libraries compatible with STL models to streamline the detection process.

Due to the complexities involved in accurate 3D shape detection—particularly with models that approximate surfaces via triangulation—we require additional research time to evaluate and implement these improvements. This will enable us to meet the project's objective of accurate, automatic shape recognition in 3D models, paving the way for reliable feature extraction and analysis capabilities in future iterations.